interface is automatically launched by the launch of the target application and runs until the target application running.

b) Ferguson interface can be used only with a Web browser for the specific purpose of downloading Web pages in the background. It has no relevant functionality in "accompaniment" with any other non-browser application, which does not carry Web links for "drag and drop" download. The instant invention runs with any and all non-browser application as well as browser application with the sole purpose of automatic, non-obtrusive display of sponsor information.

c) The Ferguson application, upon launch, will either reduce the size of the browser window or block the menus buttons in the substrate area of the browser window if running as an overlaid concurrent window. In either case it will be obtrusive to the user. The instant invention compromises neither the size of the application window, nor access to the menus / buttons.

d) The Office Action does admit that Ferguson "teaches a completely opaque "floating window" tool, and thus not one "allowing automatic cursor-responsive access to the hidden / invisible controls - -". However examiner construes that Wilks "MULTI-STATE WINDOW" discloses the "automatic cursor responsive window" of the instant invention, because the multi-state window "may exist in the two states shown in fig 1 (translucent) and 2 (opaque), but it is never completely defocused or closed". See Last Office Action, p4-5. The difference between Wilks and the instant invention is explained herein.

e) Those skilled in the art know that multiple windows ordinarily do not remain in focus simultaneously. Only one window remains in focus at a given time. Consequently, when Wilks current window is in focus "the multi-state window [ ] return[s] to the translucent, non-focused state." See Wilks, Claim 1(e). In contrast the floating window simultaneously "retains focus" even when the user is working in the current application and such floating window cannot be "unfocussed". See Claim 249 and Claim 259 ("simultaneously in active focus").

f) The translucency of Wilks "multi-state window" renders it useless for display of advertising banners or graphics or any clearly readable message. If the multi-state window is in unfocussed state, the faint image of the overlaid areas of the focused current window will blur any advertisement contained on the "multi-state window". Conversely, when it is in focus the underlying areas of the unfocussed current window will not be accessible.

g) The transition of "the non-focused state" translucent window to "focused" opaque window is not automatic, but requires "selection of the multi-state window" by one of the means in Wilks claim 5. Similarly the transition from opaque to translucent also requires selecting the current window. In the instant case the floating window

always remains in simultaneous focus, and does not require any "selection" or focusing of any of the windows to make it disappear to access the masked tools and buttons in the substrate area. The access to the overlaid substrate area is automatic without any cursor click or selection.

h) The Wilks interface is primarily designed to economize the screen space for control buttons, by placing them in overlapping layers of translucent multi-state windows, each of which can be sequentially brought into focus as needed. It does not create any space, which can non-obtrusively display sponsor advertisement information.

4. The Ferguson interface cannot become "non-obtrusive in the style of Wilks", because Wilks itself is not capable of displaying "non-obtrusive" advertisements and still allow automatic access to the masked application area without having to manually change the focus of the windows by the user. Therefore, both the independent claims 249 and 259 sustain novelty over Ferguson and Wilks. The dependant claims 257, 258 and 267, 268 are withdrawn. Although the novelty of the remaining dependant claims depend upon the two independent claims, the objections to the remaining dependant claims are nonetheless addressed herein.

5. As per claim 250 and 260, the "document area" and the "substrate area" in the Ferguson browser is pushed down by the Ferguson interface's location on the top. Therefore the document and substrate areas are reduced by the space taken by the Ferguson interface, and cannot be "60 to 80" or "20 to 40" respectively. Moreover, the Ferguson Q-links are URL addresses of the links being downloaded and not "messages" from sponsor, and they do not "comprise part or all of the substrate areas".

6. The dependant claims 251 and 261 are amended. As explained, the Ferguson interface cannot be read as "overlaid substrate controls". Even if it is construed as "overlaid substrate controls", Wilks does not create replica of the overlaid window, but the window drivers route the input message to the same overlaid window. The claims 252 and 262, which creates replicas "in any part of the document area", are therefore novel over Ferguson and Wilks.

7. The Wilks multi-state window does not provide buttons / controls to automatically swap information with the overlaid window as claimed in claim 253 and 263. Instead it allows access to the overlaid window controls only when the overlaid window is brought into focus.

8. Wilks multi-state window is integrated within the application of the "current window". It acquires no control over windows of other applications running on the client machine as claimed in claims 254 and 264.

9. Novelty of claim 255 and 265 lie in their corresponding independent claims. Merely rotational display of ads delivered from a web server per se is neither novel to Ferguson nor the instant invention.

10. Both Ferguson and Wilks teach a client-installed software application. Neither Ferguson, nor Wilks functionality can be deployed exclusively from a remote server without any installation on the client machine. Claims 256 and 266 however teach an innovation, which can be deployed from a remote server without requiring any software installation on the client machine.

11. The final claims incorporating the suggested changes and objections are herein presented for approval.

249. **(Corrected)** A method of creating one or more floating windows, by permanent default, anchored or aligned to the subtrate or deocument areas of the integrated or an unrelated software application, installed on a client machine, enabling real time continuous or intermittent display of Web-compliant sponsor data files, which window retains focus and is non-obstrusive, allowing automatic cursor-responsive access to the hidden / invisible controls, tools and any such information masked by such window; and cannot be closed, moved, manipulated, interacted or unfocused, in any other way than defined by the sponsor.

250. The method of claim 249, wherein the document area comprises about 60 to 80 percent of the application window, the substrate areas comprises about 20 to 40 percent of the application window, and the messages displayed in the substrate areas comprise part or all of the substrate areas.

251. **(Amended)** The method of claim 249, wherein the program algorithm is a Web application delivered from a Web application server as a Web page applet.

252. The method of claim 249, wherein the menus, controls, tools and any other information masked by the floating window can be accessed by launching replicas of such menus, controls, tools etc in the form of embedded Java applet or DHTML form in any part of the document area in response to a mouse or keyboard command.

253. **(Corrected)** The method of claim 249, wherein the menus, tools, controls and other information are compiled in the sponsor message window itself, which information can be swapped with the application in focus, in response to a mouse click or a keyboard strike or a combination of both, when the user desires to access such menus, tools, controls and other information.

254. The method of claim 249, wherein the floating window is programmed to display its non-obtrusive cursor-responsive functionality, in more than one or all the applications installed on the client machine.

255. **(Corrected)** The method of claim 249, wherein the floating window displays and auto updates the display, by auto retrieving substitute files from computer's storage area and / or by downloading the corresponding files from predefined web site URL using an HTTP, FTP or any Web-compatible file tranfer protocol.

256. The method of claim 249, wherein the floating window is delivered to the client machine as a Web page by an Internet server application in reponse to a specific URL request by the client.

257. (Withdrawn).

258. (Withdrawn).

259. **(Corrected)** A method of creating one or more floating windows in a GUI application running on a client machine, wherein such window:

a) is always located outside the document display or data input area of an application, but within the outer limits of such application, such area herein referred as the substrate area of the application;

b) is anchored or aligned with part or all of the substrate areas of the application;

c) is simultaneously in active focus of the same or parallel running application;

d) is displayed by permanent default with either launch of an application or is delivered by a Web server application in response to a specific URL request;

e) is displayed continuously or intermittently, for either a predefined length of time or until the closing of the application;

f) disappears, swaps or moves to another location exposing the menus, controls, tools etc. in the substrate area instantly;

g) displays contents transmitted in any Web compliant format such as HTML or Java, which content can be accessed for full view display, by a keyboard command or by keyboard mouse command combination; and

h) can neither be closed, moved, manipulated or interacted in any other way than illustrated in (a) through (g), by the user.

260. The method of claim 259, wherein the document area comprises about 60 to 80 percent of the application window, the substrate areas comprises about 20 to 40 percent of the application window, and the messages displayed in the substrate areas comprise part or all of the substrate areas.

261. **(Amended)** The method of claim 259, wherein the program algorithm is a Web application delivered from a Web application server as a Web page applet / servelet.

262. The method of claim 259, wherein the menus, controls, tools and any other information masked by the floating window can be accesssed by launching replicas of such menus, controls, tools etc in the form of embedded Java applet or DHTML form in any part of the document area in response to a mouse or keyboard command.

263. **(Corrected)** The method of claim 259, wherein the menus, tools, controls and other information is compiled in the sponsor message window itself which information can be swapped with the application in focus, in response to a mouse click or a keyboard strike or a combination of both when the user desires to access such menus, tools, controls and other information.

264. The method of claim 259, wherein the floating window is programmed to display its non-obtrusive functionality, in more than one or all the applications installed on the client machine.

265. **(Corrected)** The method of claim 259, wherein the floating window displays and auto updates the display, by auto retrieving substitute files from computer's storage area and / or by downloading the corresponding files from predefined web site URL using an HTTP, FTP or any Web-compatible file transfer protocol.

266. The method of claim 259, wherein the floating window is delivered to the client machine as a Web page by an Internet server application in response to a specific URL request by the client.

267. (Withdrawn).

268. (Withdrawn).

For all the above reasons the two independent claims 249 and 259 are distinctly novel over Ferguson and Wilks. The remaining dependent claims inherit the novelty of the invention from the independent claims. The applicant therefore respectfully submits that the application, with the changes in response to the last office action, is in allowable condition.

Respectfully submitted,

Fazal Raheman
C/o Don Mossman
30 Liberty Drive,
Blackstone, MA 01504